DOCUMENT RESUME

ED 272 171 IR 012 230

TITLE	Guideline for Software Documentation Management.
INSTITUTION	National Bureau of Standards (DOC), Washington, D.C.
REPORT NO	FIPS-PUB-105
PUB DATE	6 Jun 84
NOTE	32p.
AVAILABLE FROM	National Technical Information Service, U.S. Department of Commerce, Springfield, VA 22161.
PUB TYPE	Guides - General (050)
EDRS PRICE	MF01/PC02 Plus Postage.
DESCRIPTORS	Administrators; *Computer Software; Content Analysis; Programers; *Specifications; *Standards
IDENTIFIERS	*Printed Materials; *Software Maintenance

ABSTRACT

Designed as a basic reference for federal personnel concerned with the development, maintenance, enhancement, control, and management of computer-based systems, this manual provides a general overview of the software development process and software documentation issues so that managers can assess their own documentation requirements. Reference is then made to relevant standards, guidelines, articles, and books that can be used to develop in-house standards, guidelines, and procedures suited to the specific development environment and task. The manual also outlines factors that managers must consider if documentation is to play its proper role in the development of computer software: (1) the component parts of software documentation; (2) how managerial guidance can ensure good documentation by the implementation of policies and procedures; and (3) the resources needed to make the managerial guidance effective. A glossary is provided, as well as appendices which contain a list of document types; policy, planning, and procedures checklists for documentation; a list of nine Federal Information Processing Standards (FIPS) and Guidelines publications; a list of six additional standards and guidelines publications; and a bibliography of 15 related books and other references. (JB)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
X This document has been reproduced as received from the person or organization originating it.
☐ Minor changes have been made to improve reproduction quality.
• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

ED272171

FIPS PUB 105

FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION

1984 JUNE 6

U.S. DEPARTMENT OF COMMERCE/National Bureau of Standards

GUIDELINE FOR SOFTWARE DOCUMENTATION MANAGEMENT

CATEGORY: SOFTWARE
SUBCATEGORY: DOCUMENTATION

IR012330 FIPS PUB 105

U. S. DEPARTMENT OF COMMERCE, Malcolm Baldrige, *Secretary*
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Director*

Foreword

The Federal Information Processing Standards (FIPS) Publication Series of the National Bureau of Standards (NBS) is the official publication series relating to standards adopted and promulgated under the provisions of Public Law 89-306 (Brooks Act) and under Part 6 of Title 15, Code of Federal Regulations. These legislative and executive mandates give the Secretary of Commerce important responsibilities for improving the use and management of computers and automatic data processing in the Federal Government. To carry out the Secretary's responsibilities, the NBS, through the Institute for Computer Sciences and Technology, provides leadership, technical guidance, and coordination of Government efforts in the development of guidelines and standards for the use and management of computer technology.

The need for policies and procedures for all types of software documentation has been recognized for some time. Reports by the Comptroller General, among others, have for some time pointed out needed improvements in computer system documentation. Missing or inadequate documentation has caused severe financial losses and time delays. Many managers dealing with computer systems report similar experience. Therefore, NBS offers this Guideline to give Federal agencies information about managing the planning, development, and maintenance of adequate software documentation.

James H. Burrows, *Director*
Institute for Computer Sciences and Technology

Abstract

This Guideline can assist managers in establishing policies and procedures for effective preparation, distribution, control, and maintenance of documentation which will aid in the re-use, transfer, conversion, correction, and enhancement of computer programs. It outlines policies, procedures, and applicable standards and provides checklists in support of documentation policies, and procedures. It also includes references to relevant standards, guidelines, and the literature and a glossary of terms. Adequate software documentation, together with the computer programs themselves, provide software product packages that can be transferred and used by people other than the originators of the programs.

Key words: documentation; Federal Information Processing Standards Publication; guidelines; life cycle; software; specifications; standards.

Natl. Bur. Stand. (U.S.) Fed. Info. Process. Stand. Publ. (FIPS PUB) 105, 28 pages
(1984)
CODEN FIPPAT

For sale by the National Technical Information Service, U.S. Department of Commerce, Springfield, VA 22161

Federal Information Processing Standards Publication 105

1984 June 6

ANNOUNCING THE

GUIDELINE FOR SOFTWARE DOCUMENTATION MANAGEMENT

Federal Information Processing Standards Publications are issued by the National Bureau of Standards pursuant to the Federal Property and Administrative Services Act of 1949, as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973), and Part 6 of Title 15 Code of Federal Regulations (CFR).

Name of Guideline: Guideline for Software Documentation Management (FIPS PUB 105).

Category of Guideline: Software, Documentation.

Explanation: This Guideline provides explicit advice on managing the planning, development, and production of computer software documentation.

Approving Authority: U.S. Department of Commerce, National Bureau of Standards (Institute for Computer Sciences and Technology).

Maintenance Authority: U.S. Department of Commerce, National Bureau of Standards (Institute for Computer Sciences and Technology).

Applicability: This Guideline is a basic reference for Federal personnel concerned with development, maintenance, enhancement, control, and management of computer-based systems. The document should be used along with other applicable guides, standards, and references.

Implementation: This Guideline should be consulted early in the planning of projects that deal with development, change, or enhancement of computer programs or programming references.

Specifications: Federal Information Processing Standards Publication 105 (FIPS PUB 105), Guideline for Software Documentation (affixed).

Where to Obtain Copies of this Guideline: Copies of this publication are for sale by the National Technical Information Service, U.S. Department of Commerce, Springfield, VA 22161. When ordering, refer to Federal Information Processing Standards Publication 105 (FIPSPUB105), and title. Specify microfiche if desired. Payment may be made by check, money order, or NTIS deposit account.

ACKNOWLEDGMENTS

This Guideline is based on NBS Special Publication 500-87, *Management Guide for Software Documentation*, January 1982. The original has been extensively reviewed and updated and is now being published as a FIPS Guideline. It was prepared under contract to the National Bureau of Standards.

Federal Information
Processing Standards Publication 105

1984 June 6

Specifications for

GUIDELINE FOR SOFTWARE DOCUMENTATION MANAGEMENT

CONTENTS

		Page
1.	PURPOSE OF THIS GUIDELINE	4
2.	WHAT SOFTWARE DOCUMENTATION IS	4
2.1	Defining Software Documentation	4
2.2	What Documentation Does	4
2.2.1	Management Communications	4
2.2.2	Task-to-Task Communication	4
2.2.3	Instruction and Reference	5
2.2.4	Quality Assurance, Maintenance, and Audit Support	5
2.2.5	Historical Reference	5
2.3	What Documentation Describes	5
2.4	Development Documentation	5
2.4.1	System Life Cycle	6
2.4.2	Typical Development Documents	7
2.4.3	Purposes of Development Documents	7
2.4.4	Manager's Role in Development Documentation	7
2.5	Product Documentation	7
2.5.1	Audiences for Product Documentation	8
2.5.2	Programmer Documentation	8
2.5.3	System Administrator/Operator Information	8
2.5.4	User Training and Reference Materials	9
2.5.5	Promotional/Informational Documents	9
2.6	Defining Responsibilities for Documentation	9
2.7	Documentation Life Cycle	10
2.7.1	Documentation Quality	10
2.7.2	Presentation Formats for Documentation	11
2.8	Summary of What Software Documentation Is	11
3.	MANAGERIAL GUIDANCE	11
3.1	Documentation Policy	12
3.2	Documentation Planning	12
3.2.1	Documentation Plan	12
3.2.2	Project Librarian	13
3.2.3	Storage of Vital Documentation	13
3.2.4	Document Reviews	14
3.3	Procedures	14
3.4	Standards and Guidelines	14
3.4.1	Finding Standards and Guidelines	15
3.4.2	Using Standards and Guidelines	15
4.	REQUIRED RESOURCES	15
4.1	People	15
4.2	Facilities	15
4.3	Funding	16
5.	CONCLUSIONS	16

Page

6. GLOSSARY 16
APPENDIX 1: DOCUMENT TYPES 20
APPENDIX 2: POLICY CHECKLIST 21
APPENDIX 3: PLANNING CHECKLIST 22
APPENDIX 4: PROCEDURES CHECKLIST 23
APPENDIX 5: FIPS STANDARDS AND GUIDELINES 24
APPENDIX 6: OTHER STANDARDS AND GUIDELINES 26
APPENDIX 7: BOOKS AND OTHER REFERENCES 27

1. PURPOSE OF THIS GUIDELINE

To help managers obtain good documentation, this Guideline offers:

1. A general overview of the software development process and software documentation issues so that managers can assess their own documentation requirements.
2. References to relevant standards, guidelines, articles, and books that can be used to develop in-house standards, guidelines, and procedures suited to the specific development environment and task.

This Guideline outlines factors that managers must consider if documentation is to play its proper role in the development of computer software. This Guideline discusses:

- What software documentation is
- How managerial guidance can ensure good documentation by the implementation of policies and procedures
- What resources are required to make the managerial guidance effective.

2. WHAT SOFTWARE DOCUMENTATION IS

2.1 Defining Software Documentation

For the purpose of this document, *software documentation* is defined as:

> All information that describes the development, operation, use, and maintenance of computer software. This information is in a form that can be reproduced, distributed, updated, and referred to when it is needed.

In discussions of software documentation, it is important to specify what is meant by **documentation**. And, in the planning of documentation, it is important to look at all parts of the software development process and ensure that each part is documented. For this purpose, it is helpful to look at documentation in two ways:

- By what it does
- By what it describes

2.2 What Documentation Does

Documentation fulfills five important functions [7-6*]:

- Communications to management about the progress of the project, providing intermediate product visibility
- Task-to-task communication
- Instruction and reference
- Quality assurance support
- Historical reference

2.2.1 Management Communications

During system development, management needs to be apprised of progress, problems, and expectations. Periodic reports that track progress against schedules and lay out plans for the next period provide visibility for the project. They also remind management at all levels of the organization's commitment to the project and of its importance to the organization's effectiveness.

2.2.2 Task-to-Task Communication

Most software development projects are divided into tasks, often carried out by different groups. Experts in the subject matter initiate the project. Analysts formulate system requirements, and designers develop overall program designs for programmers, who provide detailed code. Quality assurance specialists and

*References are to sources cited in the appendices; the first number (7 in this instance) is the number of the appendix, the second (5) is the number of the source in the appendix.

auditors assess overall system integrity and performance. Maintenance programmers improve operations or develop enhancements or extensions.

These people need some way to communicate with one another that provides information that can be reproduced, distributed, and referred to as needed; thus, task-to-task communication is an important documentation function. Most system development methodologies establish formal documents for task-to-task communications. Analysts present formal requirements statements to designers; designers give formal design specifications to programmers, and so on.

2.2.3 Instruction and Reference

Another function of software documentation is to teach system administrators, operators, users, managers, and other interested persons how the system works and how to use it for their purposes. Likewise, maintenance personnel need system descriptions to help them find the source of previously undetected errors and improve the system to meet changing user requirements or changing system environments.

2.2.4 Quality Assurance, Maintenance, and Audit Support

Those charged with maintaining the system and with assessing how well the system performs require program descriptions, testing and evaluation plans, standards of quality against which to measure the system, and clear descriptions of what the system is expected to do and how it is supposed to do it. Test plans and procedures must be created and results of tests reported. Security controls, calculation and check-digit routines, and other control techniques must be described and evaluated. Such documents supply maintenance, quality assurance, and auditing personnel with the information they need to perform their tasks.

2.2.5 Historical Reference

An often-overlooked function of software documentation is its use as a resource for future projects. With technology changing so rapidly, system developers can review previous systems to find out what has been tried and what worked well and what was rejected as unworkable for some reason. Good system documentation can assist in the transfer and conversion of programs to new system environments. It can also prevent false starts by illustrating ineffective and effective solutions to software and organization problems.

2.3 What Documentation Describes

A computer-based system consists of automated and manual operations that together meet a need for information or solve a problem. Complete documentation of such a system involves describing both the development and operation of the system. For this purpose, it is helpful to divide documentation into two types:

- Development documentation—which describes the development process itself
- Product documentation—which describes the result of the development process

Figure 1 is a diagram of these two types of software documentation and shows the types of documents included in each type. Following the figure, the two documentation types are described in more detail.

2.4 Development Documentation

The documents that describe a system's development specify what users need and what the system's computer programs do. Development documentation also specifies how programs should be constructed, how they should be tested, and how their quality is to be assured.

Development documentation is closely related to the notion of a software "system life cycle." The life cycle of a software product begins when the idea for the product is formulated and ends when use of the product ends. The life cycle divides development into stages or phases, each with its own milestones. Managers use the phases to monitor progress and direct efforts where they are needed to keep the project on schedule.

Figure 1. *Major Types of Software Documentation.* The documents for each type depend on the needs and intentions of the organization; those presented in the figure are typical. Names may vary slightly with different methodologies.

2.4.1 System Life Cycle

Life cycles consist of well-defined activities. The names of activities vary from methodology to methodology, even from project to project, but they include the following concepts:

Initiation. A request for the development of a system to meet a need for information or to solve a problem for the organization making the request.

Requirements Analysis. Determination of what is required to automate the function(s) identified by the organization.

Design. Specifying the automated and manual functions and procedures, the computer programs, and data storage techniques that meet the requirements identified and the security and control techniques that assure the integrity of the system.

Programming. Coding of the program modules that implement the design.

Testing and Quality Assurance. Ensuring that the system works as intended and that it meets applicable organization standards of performance, reliability, integrity, and security.

Operation. Incorporation and continuing use of the new system by the organization.

Maintenance/Enhancement. Resolving problems not detected during testing, improving the performance of the product and modifying the system to meet changing requirements.

The names that identify the phases may vary; they are likely to be different in each organization. What is critical is that phases are specified, planned, and scheduled, and that **appropriate documentation is defined, planned for, produced, and updated** at each stage.

2.4.2 Typical Development Documents

Development documents include:

- Feasibility studies and initiation requests
- Definitions of responsibilities
- Requirements and functional specifications (what the system does)
- Design specifications, including data storage and programming specifications
- Development plans
- Schedules for each phase and records of schedule changes
- Test and implementation plans
- Quality assurance plans, standards, and schedules
- Security and control information
- Memoranda or change control forms that record agreed changes to the system as it develops. (The information in these memos should also be reflected in updated development documents.)

2.4.3 Purposes of Development Documents

Development documents serve four purposes:

1. They are the vehicle for communications between the organization's subject matter experts and the computer experts. They record the details of decisions about requirements, design, programs, and testing.
2. They delineate the responsibilities of the development team. They define who does what and when by specifying the roles of the computer experts, the organization's subject matter experts, quality assurance personnel, and anyone else involved in system development.
3. They define checkpoints and schedules that allow managers to assess the progress of the project. If they are missing, incomplete, or outdated, managers lose an important tool for tracking and controlling the project.
4. They record the history of the system's development, so that the rationale for the system's structure is available for later use.

2.4.4 Manager's Role in Development Documentation

The manager ensures that the development phases are well-defined and that roles are clearly articulated, scheduled, and documented.

Well-defined phases and tasks provide checkpoints along the way for managers. With phases named and identified, managers can monitor the progress of the data processing side of the development effort, and, at the same time, ensure that the system's development documentation is completed on time.

Commitment to development documentation is often made as part of a development methodology. These are the documents that declare intentions, set out goals, establish deadlines. If they are up-to-date and done well, they assure that all participants in the process understand where they are and where they are going.

Initial good intentions must be followed by actions like providing clerical support and insisting that appropriate documentation for one phase be completed and up to standard before the next phase can begin.

2.5 Product Documentation

While development documentation is essential as a management tool for tracking the progress of the project, product documentation provides the information necessary for the effective use, operation, maintenance, conversion, and transfer of the software system.

A **program product** or **software product** is a well-tested set of computer programs fully documented and supported by a responsible organization. The product may be commercially available, or it may be produced by a non-commercial source, but it is intended for wide application and use. It differs from an experimental or temporary program intended for private or personal use, for which documentation may be minimal because the program is short-lived and its use is limited to the developers.

If a program is to be used by other than the developers, it must be documented as a product; otherwise, a host of problems may cause the product to fail.

2.5.1 Audiences for Product Documentation

Product documentation includes materials for three groups of product users:
- Programmers who maintain or enhance the product
- Operators who run the product on a computer system
- End users who enter data or retrieve information with the product

Product documentation may also include guides and materials for managers who supervise the use of the product, promotional materials announcing the availability of the product and detailing the software and hardware requirements for use of the product, and general public relations information describing the product for those interested.

2.5.2 Programmer Documentation

Programmers charged with maintaining or enhancing an existing software program require information that describes what the program is supposed to do and when it is doing it. They need illustrations and descriptions of program logic, final data storage design specifications, and functional descriptions.

The form of the programmer's documentation is less important than its existence, but there are several usual types of program documentation:

In-line comments. Any well-written computer program contains a sufficient number of comments to permit people to read it. Development programmers should prepare these comments when they are coding and update them as the programs change.

Guidelines for this kind of program documentation often appear in programming standards. In general, each program module contains a description of the logic, the purpose, and the rationale for the module. Such comments may also include references to subroutines and descriptions of conditional processing.

Specific comments for specific lines of code may also be necessary for unusual coding. For example, an algorithm (formula) for a calculation may be preceded by a comment explaining the source of the formula, the data required, and the result of the calculation and how the result is used by the program. Ensuring that development programmers use the facilities of the programming language to interpolate comments in the code and to update those comments is an important dimension of good software development management.

Graphic Representations. Verbal descriptions of program logic often are so involved that they become impossible to follow. Diagrams or flowcharts, which depict the flow of data through a program or system, show maintenance programmers how the section of the program they are working on relates to other programs or modules in the system.

Storage specifications. Maintenance programmers must understand where data is stored and in what form. Without such basic information, their maintenance task becomes impossible.

Formal system documentation. Sometimes maintenance programmers require more than in-line comments and descriptions, graphic representations, and storage specifications. They also need written descriptions to accompany them. The more elaborate the software product, the more need there is for formal, written system or program documentation.

Clearly, some program documentation is also development documentation. The functional specifications, the data storage specifications, and so on are produced as part of the development documentation. They become part of the product documentation because they contain information that maintenance programmers need to keep the system functioning efficiently.

2.5.3 System Administration/Operator Information

System administrator/operator documents tell those in charge of running the system's hardware and software what they need to know to support the system. These documents include such information as:
- Schedules and names of batch jobs and projected run times
- Storage needs and how the storage is acquired by the programs

- Devices used
- Projected hours of online use of the system
- Troubleshooting guides

The specific operational information depends greatly on the software product itself. But the job of providing detailed information to system administrators and operators must be assigned, scheduled, completed, and kept up to date.

2.5.4 User Training and Reference Materials

Product documentation is incomplete without adequate information for users. Users need at least two kinds of information:

- Training—to help them become proficient in the use of the system quickly.
- Reference—to help them find answers to specific questions about system use.

In addition, users may want to know how what they do with the system fits into the overall workings of the organization. For this purpose, they need overviews and summaries that show how the parts of the system fit together and how what they do affects the rest of the system.

The formats of the documents that meet these information needs can vary. But, these documents should be planned, developed, and produced by the time the system is made a part of the organization's operation.

2.5.5 Promotional/Informational Documents

Promotional and informational documents serve two important purposes:

- Within the organization, they increase the acceptability of the system and ease the organization's transition into its use.
- Outside the organization, they make interested persons aware of new capabilities, increasing the potential for transfer of the system to other organizations.

There are several types of promotional or informational documents:

Management Summaries. Often, immediate supervisors have less hands-on experience than their employees when new systems are installed yet need certain detailed information. Management summaries can meet this need and ease the transition for the organization. Management summaries describe what the system does and how it aids the organization to perform its functions better.

Product Descriptions. In brief form, product descriptions summarize the features of the system and indicate the hardware/software requirements for the system.

Product Announcements—Internal. As an internal document, an announcement makes an organization aware that the sofware product is coming, gives a projected schedule for its implementation, lists the features and benefits of the system so that organization personnel can prepare for the transition.

Product Announcements—External. External product announcements whet the appetite of people outside the organization, increasing their awareness of the system's capabilities, making them want to know more.

Like other product documentation, the specific promotional and informational materials depend on the system being developed. These documents, however, should be planned and produced as part of the overall software development effort.

2.6 Defining Responsibilities for Documentation

Software product development has roles for:

- Subject matter experts from the organization to provide information about the organization's functions.
- Computer experts from either inside or outside the organization to provide data processing expertise.

System development methodologies contain generally well-defined roles for these two groups. It is important that, as part of the definition of system development activities, each group fully understand and fulfill its documentation responsibilities.

Designers and programmers are responsible for documentation that describes their tasks. The organization's subject matter experts are responsible for some documentation. The organization's subject matter experts provide information for and can produce parts of:

- Feasibility study

- Testing and quality assurance plans
- Requirements statement

They are usually responsible for:

- Plans for integrating the system into the organization's operation
- Schedules of all sorts
- User training and reference materials
- Promotional and informational materials

2.7 Documentation Life Cycle

Documentation is required during all phases of the development and operation of a software product. Its preparation should be viewed as a continuous effort covering the entire life cycle of the product. It begins with early drafts during project initiation and is updated at various reviews and changes in the development. It continues during programming, testing and operation through changes in the product caused by user feedback, changed user requirements, and changed system environments.

This documentation life cycle requires a determination of:

- What documents need to be produced
- What formats those documents should take
- When the documents should be completed
- How the documents are to be kept up to date
- Who is to produce the documents
- Who is to maintain the documents
- How the original documents and updated versions are to be distributed

2.7.1 Documentation Quality

Merely producing documentation because regulations, procedures, or a contract require it is not enough. Managers must also decide on the quality of documentation and how that quality is to be achieved and maintained.

Decisions about quality depend on available resources, the size and risk of the project, and exposure of the product both inside and outside the organization. Conscientious decisions can be made about the level of detail and formality of each document that is to be part of the overall documentation of the product.

For instance, a user's manual can be a set of typewritten pages stapled together, or a typeset booklet designed by a graphics professional, using distinct typography and extensive tables and graphs. Whether either of these or something in between should be produced depends on the project and its resources.

The quality of each document must be a conscious decision during its planning. Four levels of documentation quality can be identified [5-5], characterized by increasing formality and exposure of the document:

Minimal Information (Level 1). Level 1 documents are appropriate for single-use programs requiring less than one person-month of development effort. The documentation would include the program listing, development notes, test data, and program abstract.

Internal Document (Level 2). Level 2 documents can be used for special-purpose programs which, after careful consideration, appear to have no potential for sharing with others. In addition to the information provided for Level 1, Level 2 documentation includes liberal comments within the program listing to aid users in program setup and use. Formal documentation effort would be minimal.

Working Document (Level 3). Level 3 applies to programs to be used by several people in the same organization, or to programs that may be used in other organizations. Documents are typed but little review or editing is required beyond that required for a "working paper."

Formal Publication (Level 4). Level 4 applies to software products that are to be formally announced for general use. It is required by critical programs or by programs with repeated management applications, such as payroll. Level 4 documents conform to the editorial conventions and standards of the developing organization.

Reference [5-5] gives guidance for establishing these levels.

Considerations of quality apply to both the structure and content of the documentation. Contents can be judged by their accuracy, completeness, and clarity. Structure is determined by the order of parts and the simplicity of the overall arrangement. The four levels require increasing commitment and resources to achieve

the required level of quality and some quality assurance mechanism must be in place to ensure that the desired level of quality is achieved.

2.7.2 Presentation Formats for Documentation

Information can be presented in a variety of formats. For example, design specifications can be written on pre-existing forms; user training can be accomplished through an online training program, in classrooms, or through workbooks and tutorials.

The formats may vary from project to project and depend on the size of the project, the audiences to be addressed, the number of phases identified for management control, and a host of other factors.

Economics also impinge on the document formats. The cost of developing online tutorials may be too great. Typed and photocopied manuals are less expensive than typeset and printed ones, especially if the documents are likely to be revised frequently. Pre-designed forms for outlining user requirements and functional specifications and for detailing programming and storage specifications may provide more efficient communication among tasks than memoranda and written reports.

Managers must decide early during a project about the number of documents and the formats of those documents. FIPS PUB 38 [5-5] and FIPS PUB 64 [5-7] present a framework for making document decisions by outlining thirteen document types. In addition, other resources, such as those in Appendix 7, can aid in determining the format and organization of documents.

2.8 Summary of What Software Documentation Is

Software documentation is all the reproducible, human-usable material that describes the development of the software product and describes the result of that development so that the product can be used effectively and transferred or converted to other environments if appropriate.

Software documentation serves five important functions: **management communications, task-to-task communication, instruction and reference, quality assurance information, historical reference**.

Two major types—**development and product documentation**—fulfill these functions.

Managers should insist on clearly defined and documented responsibilities for those involved in the development process. Especially, managers should insist on clearly defined responsibilities for documentation planning, writing, and production as an integral, essential part of the development effort.

In addition, managers can aid in determining the formats (packaging) and quality of the documentation produced, with a clear understanding of the purpose and aim both of the system and of the documentation. Good documentation requires a continuing commitment from all those involved with the system.

3. MANAGERIAL GUIDANCE

Both technical and managerial guidance can provide solutions to documentation problems. Because of its fundamental importance, this document stresses managerial guidance which rests on three elements:

1. Management and staff commitment to documentation. This commitment requires recognition that formal or informal documentation is important and must be planned, written, reviewed, produced, distributed, and maintained.
2. Management support of the commitment to documentation. Managers can provide guidance and positive incentives for the staff to develop documentation, can designate staff to do the work, and can make the resources available to facilitate it.
3. Visible evidence of the commitment and support, including:
   - Policies on system and software documentation established, recorded, and published.
   - Planning of documentation as an integral part of the overall development effort.
   - Procedures established for determining documentation quality, measuring the quality, and providing the means to achieve and audit the quality.
   - Standards and guidelines identified and prepared for all aspects of documentation.
   - Organizational climate conducive to documentation work, with managers clearly supporting integration of the documentation effort into the development effort.
   - Continuous review process established to ensure compliance with policy and procedures and observance of standards and guidelines.

3.1 Documentation Policy

Prepared and supported by the highest echelon in an organization, policies guide decision-makers at all lower levels. Policy provides broad direction but not detailed prescriptions on what to do or how to do it.

Policy may be informal, unwritten, and undeclared, but formal, written, well-publicized policy clearly establishes the discipline required for high quality software documentation.

This Guideline discusses the vital role documentation plays during the planning, development, and operation of software systems. Documentation is also essential when systems are upgraded, maintained, converted, or transferred. Because of the importance of software documentation policy, some formal statements about it should be prepared, and all persons affected by the policy should be informed and aware of it.

Policies should support the basic elements of documentation:

- Documentation efforts cover the whole system life cycle. Documentation is required during the early phases of a project, must be maintained, and must be available during development. After development is completed, documentation must be available for use, maintenance, enhancement, conversion, or transfer of the programs so long as those programs are used.
- Documentation should be managed. Managers and documentation developers should prepare a detailed plan outlining documentation products, schedules, responsible persons, resources, and review and quality assurance procedures. Direction and control are required to obtain and maintain documentation.
- Documentation should be prepared for a variety of users. Users may be managers, analysts, professionals with no computer expertise, maintenance programmers, clerical personnel. Depending on tasks performed, they require various degrees of detail and different presentations of material. A documentation professional should be charged with properly designing different types of documentation destined for different users.
- The documentation effort should be integrated into the overall systems development process, and such a development process should be defined.
- Support tools should be specified which help to develop and maintain software products, including documentation, throughout the system life cycle; they should be used wherever economically feasible.
- Existing standards should be identified and used, or alternatively, a set of standards should be developed consistent with the size, scope, and exposure of the project.

The checklist in Appendix 2 provides aid for developing a policy statement or for assessing the usefulness and completeness of existing policy statements.

3.2 Documentation Planning

A documentation plan may be part of an overall project plan or a stand-alone document. The plan should be written and distributed to all development team members to serve as a concrete evidence of the importance of documentation and as a reminder of management's commitment to the documentation effort.

For small, informal projects, the plan may be only one page long. For larger projects, it may be a comprehensive, formal document that follows rigid standards with a formal review and approval procedure.

3.2.1 Documentation Plan

Planning should start early, and the plan should be reviewed throughout the project. Like any plan, a documentation plan indicates future activities and is subject to change as needs change. Regular reviews resulting in appropriate changes to the plan should be built in to the project, and the plan should be available to all persons affected by it.

A documentation plan states:

- **What** is to be done
- **How** it is to be done
- **When** it is to be done
- **Who** is to do it

In addition, the plan specifies the level of quality that each document is to reach and what external factors must be taken into account to achieve the desired results.

The plan also specifies the distribution of the plan and the documents and clearly delineates responsibilities of all those involved in the documentation effort.

Document Types and Content. A good documentation plan defines the documents to be produced and outlines the general content of each one. Appendix A lists document types and refers to sources that give guidance for their contents. Appendices 5, 6, and 7 list FIPS and related standards and guidelines.

Document Format and Identification. Standardized documentation formats are essential for maintenance and quality control. Formats can make documents easy to read, easy to use as references, and easy to maintain.

Identifying information—document numbers, revision dates or version numbers, authors, responsible organization—is essential for maintaining up-to-date documentation.

Most organizations have developed organization-wide format and identifcation standards. If no standards exist or if they are inadequate, some are required. Appendices 5, 6, and 7 list sources that can aid in developing standards and guidelines.

Schedule. The documentation plan should include a detailed schedule, listing the documents, review points, and personnel responsible for planning, writing, reviewing, and distribution.

The schedule allows time for:

- Planning the document
- Reviewing the document plan/outline
- Preparing drafts and reviewing them for technical accuracy, completeness, and appropriateness
- Editing
- Final review and approval
- Production—typesetting, printing, photocopying
- Distribution

Formal editing, typesetting, and printing are not relevant to many development documents. Planning, review, approval, production, and distribution, however, are necessary for all documents.

3.2.2 Project Librarian

On larger projects, a project librarian should be designated to collect project development data, maintain a master set of documentation, and maintain an index of project documentation. Depending on the scope and complexity of the project, the librarian's duties may be a part-time assignment for someone on the current staff, or it may be a full-time position requiring an additional person.

In addition to collecting documents and preparing an index for finding the documents, the project librarian should also maintain:

- Brief chronology of significant events
- Records of monthly estimates of machine time
- Records of monthly estimates of staff time
- List of changes to the estimates
- Summary of actual times expended

Along with the checkpoints built in to schedules and other regular reviews, such records yield information that managers need for project control.

3.2.3 Storage of Vital Documentation

Software documentation is a vital asset to an organization. It represents a significant investment in time, thought, and energy. And, it describes another significant investment—the development effort and the product.

A facility in a different location should be set up to store copies of vital documentation. This off-site storage should house backup copies of all development and product documentation. If the documents are developed online, they should be stored on tape so that they can be converted quickly to usable form.

In case of disaster—man-made or natural—backup card decks, tapes, disks, diskettes, listings, system diagrams, and so on can be used to reconstruct the system.

3.2.4 Document Reviews

In large projects, formal reviews are normally mandated by the development methodology. These reviews should include documentation to ensure that it is accurate and up-to-date. Problems can ensue if too little emphasis is placed on whether documentation is keeping pace with other aspects of development.

All documents that describe the development effort and the product should be part of the formal review process. Of particular importance initially are the reviews of the requirements and design specifications.

Requirements Reviews. The requirements reviews confirm that the developers and designers understand what the user needs and that the user understands the limitations and constraints on the developers.

Requirements reviews (and there may be more than one) results in an approved functional requirements document. Based on common understanding of what the system is to do, detailed design can begin. User representatives must participate actively in the development and review of the requirements and in approval of the requirements document.

Design Reviews. Three major design reviews are often scheduled: system design, preliminary design, and critical design.

During **system design review** the overall system structure is reviewed in light of the approved requirements. This review results in a system specification.

In a **preliminary design review,** the basic design approach and test plans for each system component are the subject of scrutiny. The system specification is modified as necessary as a result of this review.

A **critical design review** permits analysis of detailed computer program designs and initial test procedures for each program component.

Design reviews result in final documents that specify how the system and programs are to be designed, developed, and tested to meet the requirements agreed on. Formal minutes provide a record of all meetings.

These reviews—**requirements and design**—are essential regardless of the size of the project or the degree of formality in project management. Requirements must be clearly stated, and both users and developers must understand them; details of design need to be agreed on and documented to permit translation of requirements into programs and components.

Other Reviews. Formal reviews of other documentation are necessary as well. Plans for **product documentation** should include review and approval for:

- Organization
- Technical accuracy
- Completeness of coverage
- Appropriateness to the audience
- Graphics ideas and final graphics (which should also undergo separate reviews for technical accuracy, appropriateness and completeness)
- Correctness in grammar, punctuation, and other mechanics
- Adherence to format and other standards

If there are standards and guidelines (existing or developed), the documents can be judged against those standards. But formal review ensures that the product documentation is accurate, complete, and appropriate for the audience.

Appendix 3 provides a checklist of documentation planning activities.

3.3 Procedures

Procedures that support the policies outlined earlier must cover both preparation and use of documentation throughout the life cycle of the software product. Procedures suggest logical sequences for the planning, preparation, review, production, and distribution of the documents. They build in approval, quality assurance, and control points. They outline the revision process, storage and maintenance requirements, and update methods.

The checklist in Appendix 4 can help in developing appropriate procedures or in assessing the usefulness of existing procedures.

3.4 Standards and Guidelines

Standards or guidelines provide criteria for judging the completeness, usefulness, and appropriateness of

both development and product documentation. Organizations that do not already have standards and those that want to assess their current ones can find help in this section and in Appendices 5, 6, and 7.

3.4.1 Finding Standards and Guidelines

Appendix 5 lists Federal standards and guidelines published as part of the FIPS publications series by NBS. Appendix 6 lists standards and guidelines prepared by the American National Standards Institute (ANSI) and by professional societies. Appendix 7 lists books and journals that set up standards and guidelines or give suggestions about what to consider in the development of standards.

3.4.2 Using Standards and Guidelines

Whether developed in house or adopted from an existing source, most guidelines provide broad guidance that is applicable to many different situations. Managerial judgment is required to adapt the guidelines for the particular project.

Managers help determine which document types are required, how much documentation is to be provided, what the documents are to contain, what level of quality is to be achieved, and when the documents are to be produced. Thus, guidelines serve as suggestions rather than rigid specifications.

If a contract for software is let, it should require that documentation not only meet existing organization standards but should also specify the types of documents needed, the level of quality for each, and review and approval methods.

Appendices 5, 6, and 7 list guidelines that can aid in specifying the document types, levels of quality, and review and approval methods. For example, two FIPS documents—FIPS PUB 38 [5-5] and FIPS PUB 64 [5-7]—define document types and provide detailed content guides for development documentation. FIPS PUB 38 and FIPS PUB 64 are tied to a typical development life cycle representative of medium and large projects; other sources, like those in Appendices 6 and 7, are not linked to a life cycle.

These resources can provide appropriate guidance for the development of in-house documentation standards and guidelines.

4. REQUIRED RESOURCES

To develop high quality software and correspondingly high quality documentation, necessary resources must be made available. Required resources include:

- People
- Facilities
- Funding

4.1 People

Technical work and management are critically dependent on people. No guidelines, standards, or methodology can substitute for good people, making good human judgments. Some training in technical writing and documentation techniques are appropriate and useful. The prime requirement, however, remains the employment and retention of good people both for computer program development and for documentation.

4.2 Facilities

Certain automated software tools have been used successfully in the preparation of development documentation. Computer programs can provide diagrams, indexes, lists of data elements, and cross references among subroutines and other program components. Such capabilities avoid tedious retyping of draft material and permit automatic reprinting of updated documents.

Computer techniques have also been devised to check documents for consistency and to provide correlation between requirements and design documents and computer code.

Automated aids should be used if their cost and additional resources can be justified in relation to overall project resources.

4.3 Funding

Although development documentation costs are rarely identified as unique budget items and product documentation costs are often underestimated, they are a significant part of development costs.

Funds are necessary to support the writers, the facilities they use, and the storage, reproduction, distribution, and maintenance of the documents. Time and effort are required for reviews and updating. The project budget and schedules must reflect these costs.

Services of documentation specialists or other persons familiar with the field should be solicited during planning to assist in establishing a reasonable budget. Some commercial firms specialize in the production of documentation and provide a good alternative to in-house efforts. These firms often provide needed special capabilities for a limited time, and they help to identify the cost of documentation for a given project.

5. CONCLUSIONS

This Guideline identifies software documentation as a critical element in the development of computer software. If documentation is inaccurate, missing, or incomplete, the development effort is damaged, perhaps beyond repair.

Good documentation rests on two elements: better documentation techniques and managerial efforts. Once managers understand clearly what software documentation is, they can better plan, support, and control the documentation effort.

Software documentation covers the entire development life cycle and is a necessary part of a user-oriented software product. Managers commit the organization to the documentation effort and give concrete evidence of that commitment in the policies and procedures they develop.

Managers help plan the documentation, determine which document types are needed, outline the content of the documents, specify the level of quality to be achieved, set priorities for documentation preparation, support the production, distribution and update of the documents, and balance the need for documentation against needs of the overall project.

Managing the documentation of software development requires people, facilities, and funding. Managers allocate the resources and support those resources during all phases of the development project.

The general guidance in this Guideline, the appended checklists, and lists of references provide enough information to permit adequate specification of documentation requirements. These materials also can aid in the review of policies and procedures and of standards and guidelines to ensure adequacy of planning and development efforts.

6. GLOSSARY

This glossary lists some of the terms that occur frequently in software documentation. Some of the terms are from reference [5-1]. Others are based on, or adapted from, *IEEE Standard Glossary of Software Engineering Terminology* [6-6]. The terms and definitions here are examples; in many cases, major concepts may need additional clarification.

as built

Pertaining to an actual configuration of software code resulting from a software development project.

baseline

A specification or product that has been formally reviewed and agreed upon, that serves as the basis for further development, and that can be changed only through formal change control procedures.

block diagram

A diagram of a system, instrument or computer, in which the principal parts are represented by suitably annotated geometrical figures to show both the basic functions of the parts and the functional relationships between them.

build to

Pertaining to a baseline specification from which a computer program is coded.

computer program abstract

A brief description of a computer program, providing sufficient information for potential users to determine the appropriateness of the program to their needs and resources.

design specification

A specification that documents how a system is to be built. It typically includes system or component structure, algorithms, control logic, data structures, data set use information, input/output formats, interface descriptions, etc. Contrast with: **requirements specification**

development documentation

Information that describes the development of a software system. Contrast with: **product documentation**

development specification

Sometimes a synonym for requirements specification. Contrast with: **design specification**

document

A data medium and the data recorded on it that generally has permanence and is human or machine readable.

documentation

1) A collection of documents on a given subject. 2) The process of generating a document.

documentation plan

A management document describing the approach to a documentation effort. The plan typically describes what documentation types are to be prepared, what their contents are to be, when this is to be done and by whom, how it is to be done, and what the available resources and external factors affecting the desired results are.

flowchart

A graphical representation of the definition, analysis, or method of the solution of a problem, in which symbols are used to represent operations, data, flow, equipment, etc.

formal specification

1) A specification written and approved in accordance with established standards. 2) A specification expressed in a requirements specification language.

functional specification

A specification that documents the functional requirements for a system or system component. It describes what the system or component is to do rather than how it is to be built.

functional requirements document

See: **functional specification.**

interface specification

A specification that documents the interface requirements for a system or system component.

level of documentation

A description of required documentation indicating its scope, content, format, and quality. Selection of the level may be based on project, cost, intended usage, extent of effort, or other factors.

life cycle

See: **software life cycle.**

maintenance plan

A document that identifies the management and technical approach used to maintain software products. It typically includes tools, resources, and schedules.

performance specification

1) A specification that sets forth the performance requirements for a system or system component. 2) Syn. for: **requirements specification.**

programming specification

See: **design specification.**

project notebook

A central repository of written material such as memos, plans, technical reports, document drafts, etc. pertaining to a project.

project plan

A management document describing the approach taken for a project. The plan typically describes work to be done, resources required, methods to be used, the configuration management and quality assurance procedures to be followed, the schedules to be met, the project organization, etc. Project in this context is a generic term. Some projects may also need integration plans, security plans, t[illegible]ans, quality assurance plans, etc.

See also: **documentation plan, software development plan, test plan.**

quality assurance

A planned and systematic pattern of all actions necessary to provide adequate confidence that the item or product conforms to established requirements.

requirements specification

A specification that documents the requirements of a system or system component. It typically includes functional requirements, performance requirements, interface requirements, design requirements, development standards, etc.

See also: **system specification, design specification.**

software

Computer programs, procedures, rules, and possibly associated documentation and data necessary for the operation of a data processing system.

software development notebook

A collection of material pertinent to the development of a software module. Contents typically include the requirements, design, technical reports, code listings, test plans, test results, problem reports, schedules, notes, etc. for the module.

software development plan

The project plan for the development of a software product.

software documentation

Technical data or information, including computer listings and printouts, in human readable form, that describe or specify the design or details, explain the capabilities, or provide operating instructions for using the software to obtain desired results from a software system.

software life cycle

The period of time that begins when a software development project is initiated and ends when a product is no longer available for use.

software quality assurance

A planned and systematic pattern of all actions necessary to provide adequate confidence that software conforms to established requirements and standards, and that it achieves satisfactory performance.

specification

1) A document that defines requirements, details a design, or describes a product. A specification usually is the basis for contracts, awards, and agreements to "build" a product. 2) The process of developing a specification. The process includes determining and obtaining the necessary information and producing the document.

system documentation

Documentation conveying the requirements, design philosophy, design details, capabilities, limitations, and other characteristics of a system.

system life cycle

That period of time which starts when a system product is initiated and ends when a product is withdrawn from use. A software life cycle typically includes phases denoting activities such as initiation, requirements analysis, design, implementation, test, installation and checkout, operation and maintenance.

test plan

A document describing the testing that is to be performed to verify that a system or system component satisfies the specified requirements, the test personnel, and the test methods.

See also: **project plan.**

test procedure

A formal document developed from a test plan that presents detailed instructions for the setup, operation, and evaluation of the results for each defined test.

test report

A document describing the results of the testing carried out for a system or system component.

user

1) An individual applying the software to the solution of a problem, e.g., test or operation. 2) Any entity applying the software to the solution of a problem, e.g., a personnel department, another computer program, a network, an operator.

user documentation

Documentation conveying to the end-user of a system instructions for using the system to obtain desired results, e.g., a user's manual.

APPENDIX 1: DOCUMENT TYPES

The following document types are outlined in the documents referenced. The references provide detailed outlines of document content. Some guidelines also indicate how to tailor contents to suit individual project constraints (FIPS PUB PUB 38 and FIPS PUB PUB 64).

DOCUMENT TYPE	REFERENCE
Analysts Manual (Computer Models)	NBS SP 500-73
Computer Program Abstract	ANSI X3.88-1980
(See also: Software Summary)	
Cost Benefit Analysis	FIPS PUB 64
Data Base Specification	FIPS PUB 38
Data Requirements Document	FIPS PUB 38
Feasibility Study	FIPS PUB 64
Functional Requirements Document	FIPS PUB 38
Maintenance Manual	FIPS PUB 38
Maintenance Manual (Computer Models)	NBS SP 500-73
Operator's Manual	FIPS PUB 38
Operator's Manual (Computer Models)	NBS SP 500-73
Program Specification	FIPS PUB 38
Project Request	FIPS PUB 64
Quality Assurance Plan	ANSI/IEEE 730
Software Summary	FIPS PUB 30
(See also: Computer Program Abstract)	
Test Case Specification	IEEE 829
Test Design Specification	IEEE 829
Test Incident Report	IEEE 829
Test Item Transmittal Report	IEEE 829
Test Log	IEEE 829
Test Plan	FIPS PUB 38
	IEEE 829
Test Procedure Specificatio	IEEE 829
Test Report	FIPS PUB 38
Test Summary Report	IEEE 829
User's Manual	FIPS PUB 38
User's Manual (Computer Models)	NBS SP 500-73

APPENDIX 2: POLICY CHECKLIST

[ ] Has a decision been made to provide adequate documentation?

[ ] Has a policy statement dealing with documentation been published?

[ ] Has a person or organization been charged with responsiblity for the preparation of development and product documentation?

[ ] Have resources been made available for documentation?

[ ] Has a person or organization been charged with responsibility for documentation quality?

[ ] Have relationships been established among various levels of management and organization components such as software engineering, hardware engineering, system engineering, quality assurance, and documentation to identify responsibilities, required activities, and communications channels for the preparation, distribution and maintenance of documentation?

[ ] Have all documentation requirements been integrated with the overall project development schedule?

[ ] Have appropriate documentation standards been identified?

[ ] Has a position been taken with regard to support tools and automated documentation support?

APPENDIX 3: PLANNING CHECKLIST

[ ] Has a documentation plan been prepared?

[ ] Have the required document types been defined?

[ ] Have required contents been outlined and described?

[ ] Have documentation standards been identified?

[ ] Have documentation standards been developed?

[ ] Have responsibilities been assigned for:
- [ ] Document preparation
- [ ] Project librarian
- [ ] Alternate document storage
- [ ] Documentation review

[ ] Have quality criteria been established?

[ ] Have schedules been established for deliverables:
- [ ] Draft outline
- [ ] First draft
- [ ] Revised drafts
- [ ] Graphics

[ ] Have review dates been specified?

[ ] Has an approval cycle been established?

[ ] Have production methods been decided on and planned for?

APPENDIX 4: PROCEDURES CHECKLIST

[ ] Has a review procedure been established?

[ ] Has participation of analysts, developers, programmers, maintenance persons, auditors, users, and managers been considered?

[ ] Has an approval cycle been set up?

[ ] Has a distribution list been established for each document or document type?

[ ] Has a method been established for keeping documentation up to date?

[ ] Has a feedback mechanism been established to obtain user comments and reactions to documentation?

[ ] Have maintenance procedures been established for storage and distribution?

[ ] Have procedures been set up for document identification and control?

[ ] Has a facility been set up for vital document storage?

APPENDIX 5: FIPS STANDARDS AND GUIDELINES

This appendix lists Federal Information Processing Standards (FIPS) and Guidelines which have been issued by the National Bureau of Standards (NBS). Copies of these publications are available from:

National Technical Information Service
U.S. Department of Commerce
Springfield, VA 22161

Information concerning prices and related standards or guidelines may be obtained from:

Standards Processing Coordinator (ADP)
Institute for Computer Sciences and Technology
National Bureau of Standards
Gaithersburg, MD 20899

1. FIPS PUB 11-2
Guideline: American National Dictionary for Information Processing Systems
1983 May 9

Alphabetic listing of over 4000 terms and their definitions; prepared by ANSI Technical Committee X3K5, it also contains terms approved by the International Organization for Standardization.

2. FIPS PUB 20
Guidelines for Describing Information Interchange Formats
1972 March 1

Identifies characteristics of formatted information that must be considered for interchange of computer information; objective is to clarify and improve documentation for formatted information transfer; guidelines describe physical and logical characteristics; includes a glossary of terms.

3. FIPS PUB 24
Flowchart Symbols and their Usage in Information Processing
1973 June 30

Specifies standard flowchart symbols and their use; also known as ANSI X3.5-1970.

4. FIPS PUB 30
Software Summary for Describing Computer Programs and Automated Data Systems
1976 February 15

Defines a standard software summary form (SF-185) and gives instructions for describing computer programs for identification, reference, and dissemination; form used to record summaries of programs developed or acquired by Federal agencies and by GSA to register selected Government software.

5. FIPS PUB 38
Guidelines for Documentation of Computer Programs and Automated Data Systems
1974 June 30

Gives guidance for determining content and extent of 10 document types used in program and systems development, covering the development phase of the software life cycle; document types include functional requirements, data requirements, system specification, data base specification, test plan, test analysis report, and user operations, and program manuals.

6. FIPS PUB 44
COBOL Coding Form
1976 September 1

Provides a standard COBOL coding form (SF-268), with an explanation of its use and physical specifications; form used in coding of source programs or as in input document in transcription of COBOL source programs to a medium acceptable to computer systems.

7. **FIPS PUB 64**
Guidelines for Documentation of Computer Programs and Automated Data Systems for the Initiation Phase
1979 August 1

Provides guidelines for determining the content and extent of documentation during the initiation phase; covers project request, feasibility study, and cost-benefit study.

8. **FIPS PUB 99**
Guideline: A Framework for the Evaluation and Comparison of Software Development Tools
1983 March 31

9. **FIPS PUB 101**
Guideline for Lifecycle Validation, Verification, and Testing of Computer Software
1983 June 6

Intended for managers and implementors of software development projects, this Guideline recommends that validation, verification, and testing be done throughout the software life cycle. It explains how to plan for specific validation, verification, and testing requirements.

APPENDIX 6: OTHER STANDARDS AND GUIDELINES

This appendix lists standards and guidelines published by the American National Standards Institute (ANSI) and by professional organizations. Information on availability and costs can be obtained from the publishers.

1. *American National Standard Guidelines for the Documentation of Digital Computer Programs*
ANSI N413-1974

American Nuclear Society
555 North Kensington Avenue
LaGrange Park, IL 60525

2. *Guide for Technical Documentation of Computer Projects*
ANSI X3TR-6-82
Technical Report No. 3

3. *American National Standard for Computer Program Abstracts*
ANSI X3K7 X3.88-1980

For references 2 and 3:

X3 Secretariat/CBEMA
311 First Street, NW
Washington, D. C. 20001

4. *IEEE Standard for Software Quality Assurance Plans*
ANSI/IEEE Std 730
September 1981

5. *ANSI/IEEE Standard for Software Test Documentation*
ANSI/IEEE Std 829-1983
February 1983

6. *ANSI/IEEE Standard Glossary of Software Engineering Terminology*
ANSI/IEEE Std 729-1983
February 1983

For references 4, 5, and 6:

The Institute of Electrical and Electronic Engineers
345 East 47th Street
New York, NY 10017

APPENDIX 7: BOOKS AND OTHER REFERENCES

1. A. J. Neumann
Management Guide for Software Documentation
NBS Special Publication 500-87
January 1982

Institute for Computer Sciences and Technology
National Bureau of Standards
Gaithersburg, MD 20899

2. D. Walsh
A Guide to Software Documentation
1969

Advanced Computer Techniques Corporation
437 Madison Avenue
New York, New York 10022

3. M. Gray and K. London
Documentation Standards
1969

Brandon Systems Press, Inc.
New York, NY

4. M. L. Rubin
Documentation Standards and Procedures for On-line Systems
1979

Van Nostrand Reinhold Company
New York, NY

5. K. London
Documentation Standards (rev. edition)
1973

Auerbach
Philadelphia, PA

6. K. R. London
"*Documentation*" in Encyclopedia of Computer Science
edited by A. Ralston
1976

Van Nostrand Reinhold Company
New York, NY

7. R. C. Tausworthe
Standardized Development of Computer Software, Part II: Standards
1979

Prentice-Hall, Inc.
Englewood Cliffs, NJ 07632

8. *Computer Model Documentation Guide*
NBS Special Publication 500-73
January 1981

Institute for Computer Sciences and Technology
National Bureau of Standards
Gaithersburg, MD 20899

9. *Proceedings of the NBS FIPS Software Documentation Workshop*
NBS Special Publication 500-94
October 1982

National Bureau of Standards
Gaithersburg, MD 20899

10. Sandra Pakin & Associates, Inc.
DDM: The Documentation Development Methodology
1982

Sandra Pakin & Associates, Inc.
6007 North Sheridan Road
Chicago, IL 60660

11. Association for Computing Machinery
Special Interest Group for Systems Documentation (SIGDOC)
ASTERISK *
Bi-monthly

Association for Computing Machinery
11 West 42nd Street
New York, NY 10036

12. J. Van Duyn
The DP Professional's Guide to Writing Effective Technical Communication
1982

Wiley Interscience
New York, NY

13. Susan J. Grimm
How to Write Computer Manuals for Users
1982

14. William D. Skees
Writing Handbook for Computer Professionals
1982

For references 13 and 14:
Lifetime Learning Publications
Belmont, CA

15. William L. Harper
Data Processing Documentation: Standards, Procedures and Applications
1980

Prentice-Hall, Inc.
Englewood Cliffs, NJ

U.S. GOVERNMENT PRINTING OFFICE 1984 455-910 20130

NBS Technical Publications

Periodicals

Journal of Research—The Journal of Research of the National Bureau of Standards reports NBS research and development in those disciplines of the physical and engineering sciences in which the Bureau is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Bureau's technical and scientific programs. As a special service to subscribers each issue contains complete citations to all recent Bureau publications in both NBS and non-NBS media. Issued six times a year.

Nonperiodicals

Monographs—Major contributions to the technical literature on various subjects related to the Bureau's scientific and technical activities.

Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications—Include proceedings of conferences sponsored by NBS, NBS annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

Applied Mathematics Series—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a worldwide program coordinated by NBS under the authority of the National Standard Data Act (Public Law 90-396). NOTE: The Journal of Physical and Chemical Reference Data (JPCRD) is published quarterly for NBS by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements are available from ACS, 1155 Sixteenth St., NW, Washington, DC 20056.

Building Science Series—Disseminates technical information developed at the Bureau on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NBS under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NBS administers this program as a supplement to the activities of the private sector standardizing organizations.

Consumer Information Series—Practical information, based on NBS research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

Order the **above** *NBS publications from: Superintendent of Documents, Government Printing Office, Washington, DC 20402.*

Order the **following** *NBS publications—FIPS and NBSIR's—from the National Technical Information Service, Springfield, VA 22161.*

Federal Information Processing Standards Publications (FIPS PUB)—Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NBS pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

NBS Interagency Reports (NBSIR)—A special series of interim or final reports on work performed by NBS for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Service, Springfield, VA 22161, in paper copy or microfiche form.